

PDF Generator

```
// install PdfKit  
npm i pdfkit
```

```
// Create File pdfkit.js
```

```
/*
```

Simple tables with node and pdfkit

<https://www.andronio.me/2017/09/02/pdfkit-tables/>

```
*/
```

```
// Import dependencies
```

```
const PDFDocument = require("pdfkit");
```

```
class PDFDocumentWithTables extends PDFDocument {  
  constructor(options) {  
    super(options);  
  }  
}
```

```
table(table, arg0, arg1, arg2) {  
  let startX = this.page.margins.left,  
    startY = this.y;
```

```
  let options = {};
```

```
  if (typeof arg0 === "number" && typeof arg1 === "number") {  
    startX = arg0;
```

```
    startY = arg1;
```

```
    if (typeof arg2 === "object") options = arg2;
```

```
} else if (typeof arg0 === "object") {
  options = arg0;
}

const columnCount = table.headers.length;

const columnSpacing = options.columnSpacing || 15;

const rowSpacing = options.rowSpacing || 5;

const usableWidth =
  options.width ||
  this.page.width - this.page.margins.left - this.page.margins.right;

const prepareHeader = options.prepareHeader || (() => {});

const prepareRow = options.prepareRow || (() => {});

const computeRowHeight = (row) => {
  let result = 0;

  row.forEach((cell) => {
    const cellHeight = this.heightOfString(cell, {
      width: columnWidth,

      align: "left",
    });

    result = Math.max(result, cellHeight);
  });

  return result + rowSpacing;
};

const columnContainerWidth = usableWidth / columnCount;

const columnWidth = columnContainerWidth - columnSpacing;

const maxY = this.page.height - this.page.margins.bottom;

let rowBottomY = 0;

this.on("pageAdded", () => {
  startY = this.page.margins.top;

  rowBottomY = 0;
```

```

});

// Allow the user to override style for headers

prepareHeader();

// Check to have enough room for header and first rows

if (startY + 3 * computeRowHeight(table.headers) > maxY) this.addPage();

// Print all headers

table.headers.forEach((header, i) => {
  this.font("Courier-Bold")
    .fontSize(10)
    .text(header, startX + i * columnContainerWidth, startY, {
      width: columnWidth,

      align: "left",
    });
});

// Refresh the y coordinate of the bottom of the headers row

rowBottomY = Math.max(startY + computeRowHeight(table.headers),
rowBottomY);

// Separation line between headers and rows

this.moveTo(startX, rowBottomY - rowSpacing * 0.5)

  .lineTo(startX + usableWidth, rowBottomY - rowSpacing * 0.5)

  .lineWidth(2)

  .stroke();

table.rows.forEach((row, i) => {
  const rowHeight = computeRowHeight(row);

  // Switch to next page if we cannot go any further because the space is
  over.

  // For safety, consider 3 rows margin instead of just one

  if (startY + 3 * rowHeight < maxY) startY = rowBottomY + rowSpacing;

```

```

else this.addPage();

// Allow the user to override style for rows

prepareRow(row, i);

// Print all cells of the current row

row.forEach((cell, i) => {
  this.text(cell, startX + i * columnContainerWidth, startY, {
    width: columnWidth,

    align: "left",
  });
});

// Refresh the y coordinate of the bottom of this row

rowBottomY = Math.max(startY + rowHeight, rowBottomY);

// Separation line between rows

this.moveTo(startX, rowBottomY - rowSpacing * 0.3)

  .lineTo(startX + usableWidth, rowBottomY - rowSpacing * 0.3)

  .lineWidth(1)

  .opacity(0.7)

  .stroke()

  .opacity(1); // Reset opacity after drawing the line
});

this.x = startX;

this.moveDown();

return this;
}
}

module.exports = PDFDocumentWithTables;

```

```

//import pdf kit to app.js
const fs=require('fs');
const PDFDocument = require("./pdfkit");


app.get("/matches/generateFile/pdf", (req, res) => {
  Match.find((err, docs) => {
    if (err) {
      console.log("ERROR");
    } else {
      // Create The PDF document

      const doc = new PDFDocument();

      // Pipe the PDF into a patient's file

      doc.pipe(fs.createWriteStream(`backend/pdfs/matches.pdf`));

      // Add the header - https://pspdfkit.com/blog/2019/generate-invoices-pdfkit-node/

      doc
        .image("backend/images/logo.png", 50, 45, { width: 50 })

        .fillColor("#444444")

        .fontSize(20)

        .text("Here All Matches", 110, 57)

        .fontSize(10)

        .text("Imm Yasmine Tower", 200, 65, { align: "right" })

        .text("Centre Urbain Nord", 200, 80, { align: "right" })

        .moveDown();

      // Create the table - https://www.andronio.me/2017/09/02/pdfkit-tables/

```

```

const table = {
  headers: [
    "Name",
    "Address",
    "Phone",
    "Birthday",
    "Email Address",
    "Blood Type",
    "Height",
    "Weight",
  ],

  rows: [],
};

// Add the patients to the table

for (const match of docs) {
  table.rows.push([
    match.teamOne,
    match.teamTwo,
    match.scoreOne,
    match.scoreTwo,
  ]);
}

// Draw the table

doc.moveDown().table(table, 10, 125, { width: 590 });

// Finalize the PDF and end the stream

doc.end();
res.status(200).json({
  message: "HERE PDF",
});
}
});
});

```